

Smarı Restaurant Management System

SRMS

Meal-Map



Table Of Contents

01.

System
Concept &
Vision

02.

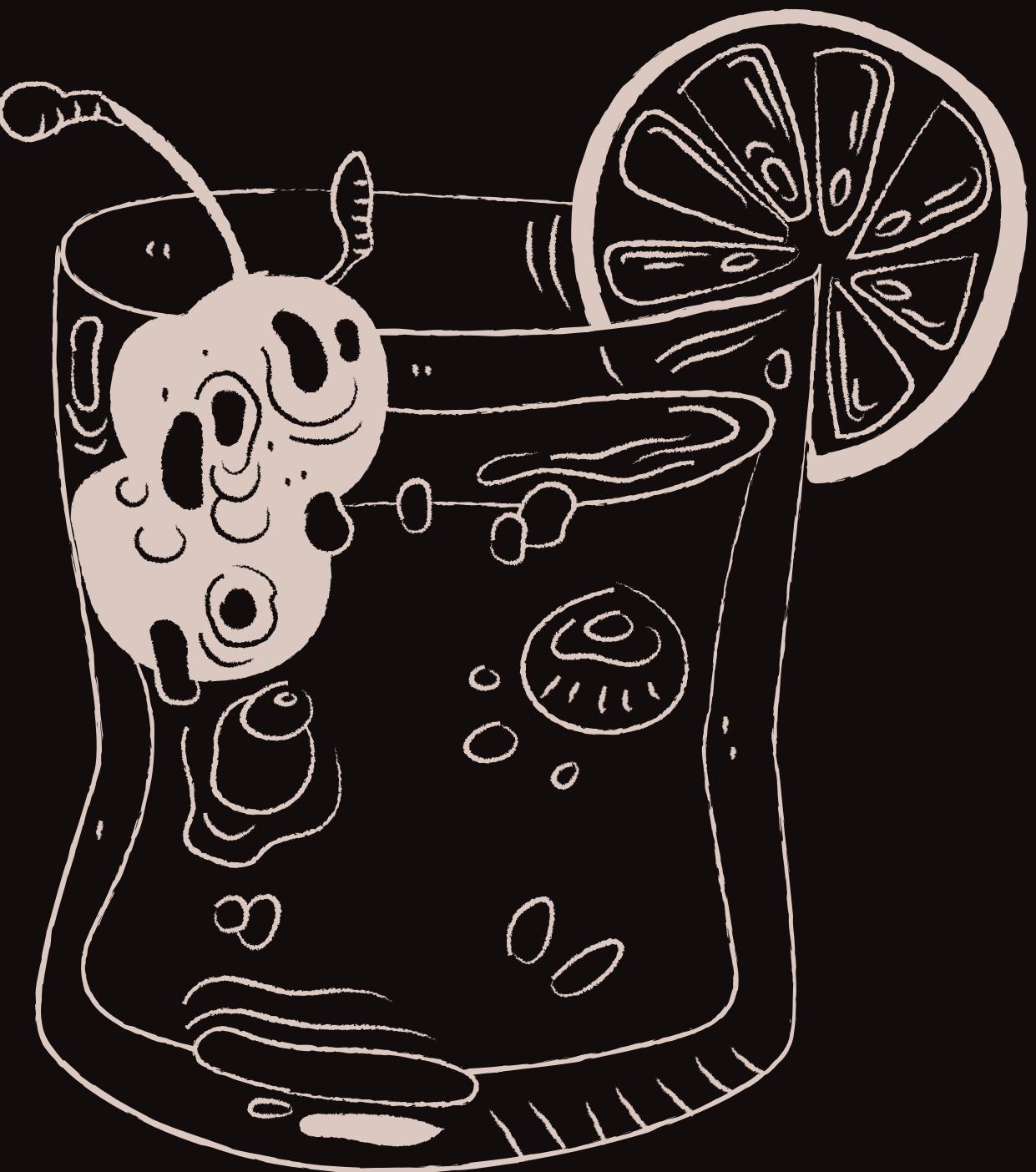
Architecture
&Core Feature

03.

RS & Technical
Implementations

01.

System Concept & Vision



What is Map-Meal?

A smart platform that connects customers and restaurant owners through an online reservation, recommendation, and management system.

Key Components:

- User App
- Owner Dashboard
- Recommender System (KB-RS)
- MySQL Hosted Online Database
- Python Backend + HTML/CSS/JS Frontend

System Concept



System Vision

The vision:

- Simplify restaurant discovery
- Provide personalized recommendations
- Enable easy table reservations
- Allow owners to manage menus, promotions & bookings
- Improve customer experience with automation



1. User / Customer:

- Create an account
- Explore restaurants
- Use recommendations
- Book tables
- Manage favourites, reviews, dashboard

2. Restaurant Owner:

- Manage restaurant info
- Manage reservations
- Update menu & promotions
- Monitor customer activity

System User



- Login (User / Owner)
- Authentication + OTP Verification
- Explore Restaurants
- Filter Preferences (Cuisine, Rating, Budget, Location)
- KB-RS recommends restaurants
- Restaurant Details
- Reservation / Reviews / Favourites
- Owner Dashboard (Menu, Promotions, Reservation Management)

High Level Project Flow



Security Features

- Password hashing
- OTP verification
- Input validation
- SQL injection prevention
- Session management



Frontend: Browser

Backend: Python Server

Database: Cloud-hosted MySQL

at AIVEN

Hosting & Deployment





02.

Architecture & Core Features



System Architecture



Presentation Layer

HTML,CSS, JAVASCRIPT



Application Layer

Python Backend Logics



Database Layer

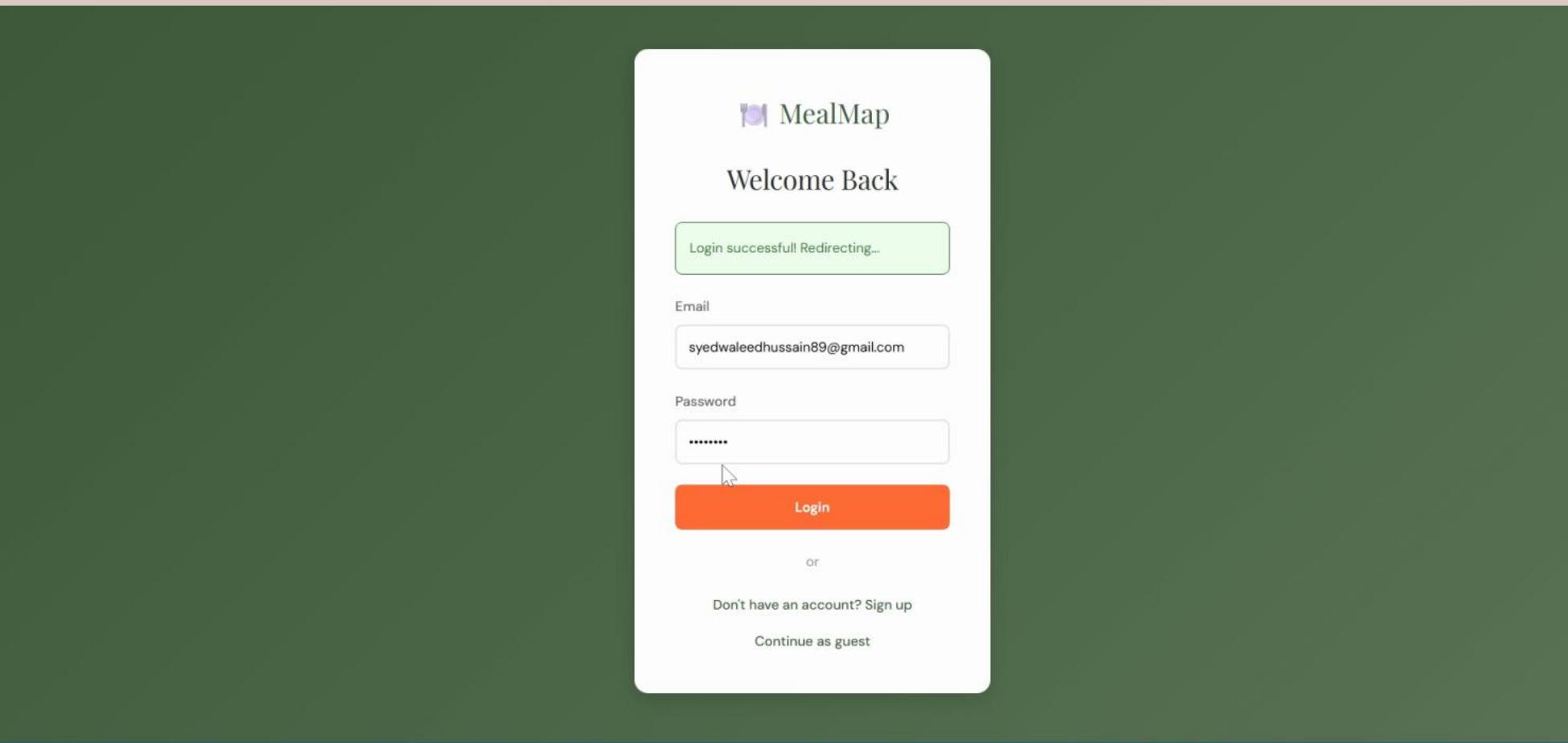
MySQL Cloud Hosted Database
on AIVEN



RS Logic Layer

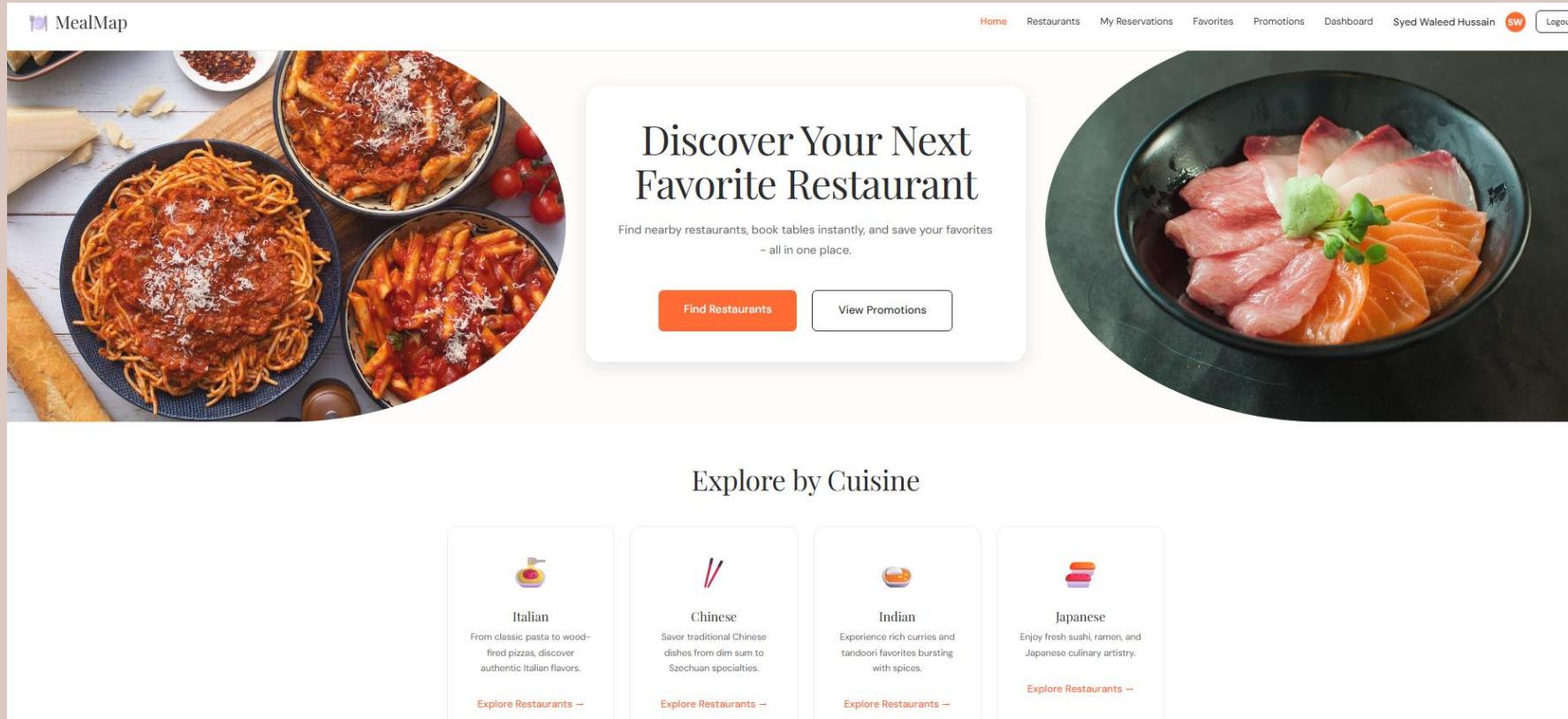
KB-RS Constraints engine –
filtering logic

Frontend UI Overview for User

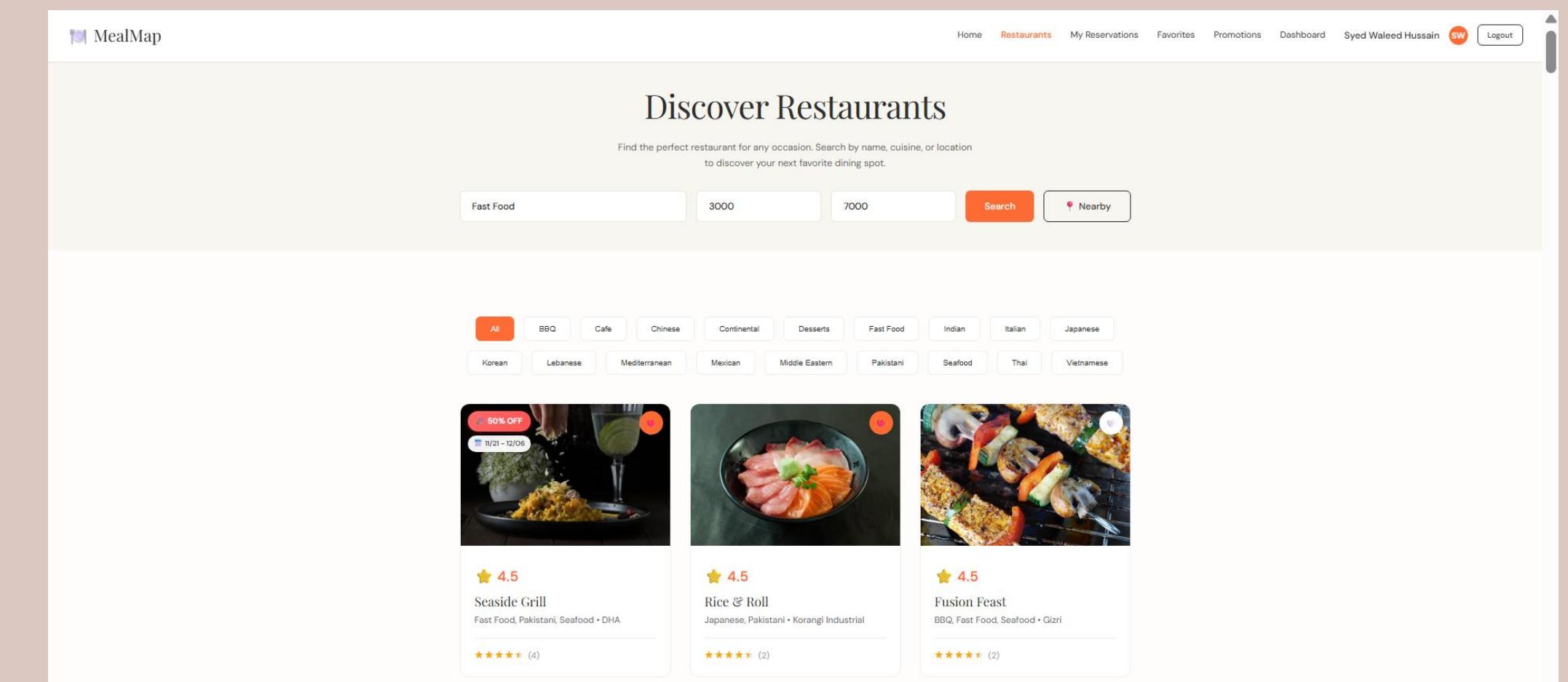


The image shows the MealMap homepage. At the top, there's a navigation bar with links for Home, Restaurants, My Reservations, Favorites, Promotions, Dashboard, and a user profile for "Syed Waleed Hussain". On the far right is a "Logout" button. The main content area features a large image of three bowls of pasta (spaghetti, penne, and ziti) with tomato sauce and cheese. To the right of this image is a white callout box with the text "Discover Your Next Favorite Restaurant" and a subtext "Find nearby restaurants, book tables instantly, and save your favorites - all in one place." It contains two buttons: "Find Restaurants" (orange) and "View Promotions" (white). Further down, there's a section titled "Explore by Cuisine" with four cards: "Italian" (with a pizza icon), "Chinese" (with a dumpling icon), "Indian" (with a curry icon), and "Japanese" (with a sushi icon). Each card has a "Explore Restaurants" link at the bottom.

Frontend UI Overview for User



The screenshot shows the MealMap homepage. At the top, there's a navigation bar with links for Home, Restaurants, My Reservations, Favorites, Promotions, Dashboard, and a user profile for Syed Waleed Hussain. A Logout button is also present. The main content area features a large image of various pasta dishes. To the right of the image is a call-to-action box with the heading "Discover Your Next Favorite Restaurant" and a subtext: "Find nearby restaurants, book tables instantly, and save your favorites - all in one place." It includes two buttons: "Find Restaurants" (orange) and "View Promotions". Below this is a section titled "Explore by Cuisine" with four categories: Italian, Chinese, Indian, and Japanese, each with a small icon and a brief description.



This screenshot shows the "Discover Restaurants" page. The top navigation bar is identical to the homepage. The main heading is "Discover Restaurants" with a subtext: "Find the perfect restaurant for any occasion. Search by name, cuisine, or location to discover your next favorite dining spot." Below this is a search bar with fields for "Fast Food", "3000", "7000", a "Search" button, and a "Nearby" button. A grid of restaurant cards is displayed, each showing a thumbnail image, the restaurant name, its rating (4.5 stars), and its cuisine type. The cards for Seaside Grill, Rice & Roll, and Fusion Feast are shown in more detail, including their addresses and review counts.

Frontend UI Overview for User

The screenshot shows a restaurant profile for "Rice & Roll". At the top, there's a large image of a dish. Below it, the restaurant's name "Rice & Roll" is displayed in a bold, black font. To the right of the name are three small icons: a star rating (4 stars), a location pin (Japanese, Pakistani), and a location pin (Korangi Industrial). A red button labeled "Remove from Favorites" is visible. At the very top, there's a navigation bar with links: Home, Restaurants, My Reservations, Favorites, Promotions, Dashboard, Syed Waleed Hussain, and Logout.

Book a Table

Date: 11/12/2025 Time: 18:32

Number of Guests: 4 Guests

Special Requests (Optional): Birthday Party

Reserve Table

The screenshot shows the "Menu" and "Reviews & Ratings" sections for the restaurant.

Menu

- Chicken Karahi**
Spicy chicken cooked in traditional wok with tomatoes and spices.
Rs. 11.50
- Sushi Platter**
Assorted sushi rolls with fresh fish and vegetables.
Rs. 14.50

Reviews & Ratings

Write a Review

Your Rating:

Your Review (optional): Share your experience...

Submit Review

Reviews

Sara Khalid (SK) - 5 stars - 07/06/2025
Loved the dessert.

Ahmed Nadeem (AN) - 5 stars - 06/06/2025
Highly recommend this place.

Frontend UI Overview for User

MealMap

Home Restaurants My Reservations Favorites Promotions Dashboard Syed Waleed Hussain SW Logout

My Favorite Restaurants

Your saved restaurants in one place



Seaside Grill
Fast Food, Pakistani, Seafood
📍 DHA
⭐ 4.5 (4 reviews)

[Remove from Favorites](#)



Crave Cafe
Italian, Mexican
📍 Malir
⭐ 4.0 (1 reviews)

[Remove from Favorites](#)



Rice & Roll
Japanese, Pakistani
📍 Korangi Industrial
⭐ 4.5 (2 reviews)

[Remove from Favorites](#)

MealMap

Home Restaurants My Reservations Favorites Promotions Dashboard Syed Waleed Hussain SW Logout

Special Offers & Promotions

Save money on your next dining experience



50.00% OFF

Azadi Discount
Flat 50% on Azadi Week
📍 Seaside Grill
Valid: Nov 21, 2025 – Dec 6, 2025

Frontend UI Overview for User

MealMap

Home Restaurants My Reservations Favorites Promotions **Dashboard** Syed Waleed Hussain SW Logout

Dashboard

 Syed Waleed Hussain
syedwaleedhussain89@gmail.com

 0 Total Reservations

 3 Favorite Restaurants

 0 Reviews Written

 0 Upcoming Reservations

Quick Actions

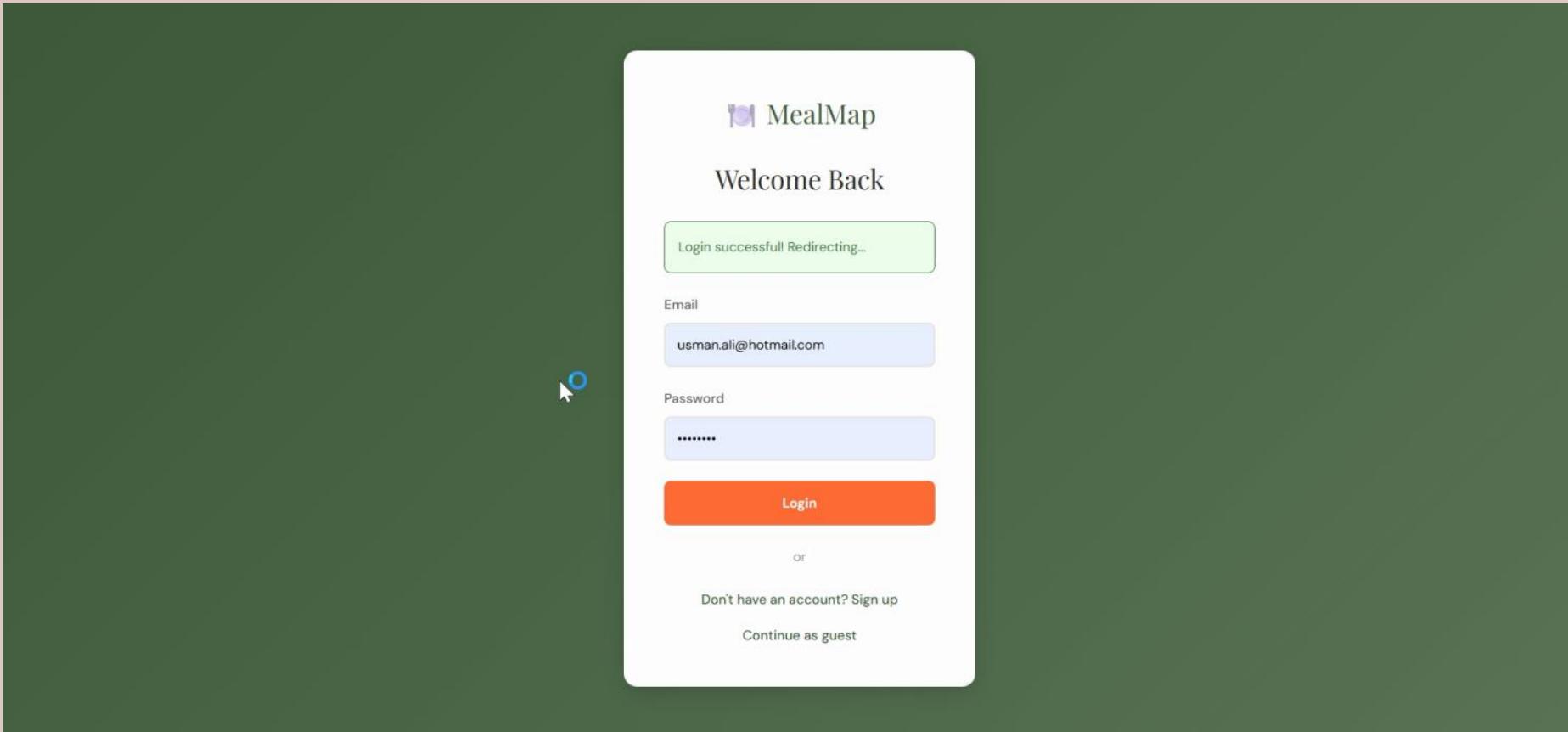
 Browse Restaurants

 My Reservations

 Favorites

 View Promotions

Frontend UI Overview for Owner



The image shows the MealMap main dashboard. At the top, there is a navigation bar with links: Home, Restaurants, My Reservations, Favorites, Promotions, Owner Dashboard, Dashboard, Usman Ali, and Logout. The "Logout" button has a red circular badge with the letters "UA". Below the navigation, there is a grid of restaurant cards. Each card includes a thumbnail image, a rating (e.g., ★ 4.0, ★ New, ★ 4.5), the restaurant name, its cuisine type, location, and distance. The cards for "The Noodle House", "Pasta Fresca", and "Ocean Breeze" are fully visible, while others are partially shown at the bottom.

Restaurant	Cuisine Type	Rating	Location	Distance
The Noodle House	BBQ, Chinese, Fast Food	★ 4.0	Bahadurabad	0.7 km
Pasta Fresca	BBQ, Japanese, Pakistani	★ New	Bahadurabad	1.1 km
Ocean Breeze	Mediterranean	★ 4.5	Bahadurabad Market	1.2 km
(Thumbnail)	(Thumbnail)	(Thumbnail)		

Frontend UI Overview for Owner

Manage your restaurant, reservations, and menu

Select Restaurant:

Tea Terrace

Restaurant Info Reservations Menu Management Promotions

Restaurant Information

Restaurant Name
Tea Terrace

Description
|

Location
Bahadurabad Block 2

Cuisine
Cafe, Continental, Fast Food

Rating

MealMap

Home Restaurants My Reservations Favorites Promotions **Owner Dashboard** Dashboard Usman Ali UA Logout

Owner Dashboard

Manage your restaurant, reservations, and menu

Select Restaurant:

Tea Terrace

Restaurant Info **Reservations** Menu Management Promotions

Reservations

All Pending Confirmed Cancelled

Date & Time	Customer	Contact	Guests	Table	Status	Actions
20/12/2025, 17:00:00	Ali Raza	ali.raza@gmail.com +923001234567	7	Table 1	Confirmed	Cancel

Frontend UI Overview for Owner

The screenshot shows the MealMap Owner Dashboard. At the top, there is a navigation bar with links for Home, Restaurants, My Reservations, Favorites, Promotions, Owner Dashboard (which is the active tab), Dashboard, Usman Ali, and Logout. Below the navigation bar, the title "Owner Dashboard" is displayed, followed by the subtitle "Manage your restaurant, reservations, and menu". A dropdown menu titled "Select Restaurant" shows "Tea Terrace" as the current selection. The main content area is titled "Menu Items" and contains three menu items: "Cheeseburger" (\$8.99), "French Fries" (\$3.50), and "Latte" (\$4.00). Each item has a description, an "Edit" button, a "Delete" button, and a toggle switch. An "Add Menu Item" button is located at the bottom right of this section.

The screenshot shows the MealMap Owner Dashboard with the "Add Menu Item" modal open. The modal has fields for "Item Name", "Description", "Price", "Image URL", and a "Available" checkbox. The "Available" checkbox is checked. At the bottom of the modal is a "Save Menu Item" button. The background shows the same dashboard structure as the first screenshot, including the "Owner Dashboard" title and the "Menu Items" section with existing menu items.



Lei's
Tasie
Backend
Now



Backend Technology:
Python

Responsibilities:

- Authentication
- OTP generation
- Hashing password
- RS filtering engine
- Reservation logic
- Favourites, reviews, promotions
- Database CRUD operations

Folder Setup

```
smart-restaurant-finder-app
├── .vscode
└── backend
    ├── __pycache__
    ├── migrations
    └── models
        ├── __pycache__
        ├── check.py
        ├── favorite.py
        ├── menu.py
        ├── photo.py
        ├── promotion.py
        ├── rating.py
        ├── recommender.py
        ├── reservation.py
        ├── restaurant.py
        └── useraccount.py
```

```
routes
├── __pycache__
├── favorites.py
├── menu.py
├── owners.py
├── photos.py
├── promotions.py
├── ratings.py
├── recommend.py
├── reservations.py
├── restaurants.py
└── users.py

utils
├── __pycache__
├── check.py
├── email.py
├── map.py
├── app.py
├── config.py
└── OWNER_SETUP_INSTRUCTI...
└── requirements.txt
```

```
database
├── smart_restaurant_finder.sql
└── smart-restaurant-finder-30-...

frontend
├── css
├── images
├── js
├── dashboard.html
├── favorites.html
├── index.html
├── login.html
├── my-reservations.html
├── owner-dashboard.html
├── promotions.html
├── restaurant-detail.html
├── restaurants.html
├── signup.html
└── verify-otp.html

.gitignore
IMPLEMENTATION_PLAN.md
main.py
server.py
```

App.py

```
3  from flask import Flask
4  from flask_cors import CORS
5  from routes.restaurants import restaurants_bp
6  from routes.users import users_bp
7  from routes.reservations import reservations_bp
8  from routes.recommend import recommend_bp
9  from routes.menu import menu_bp
10 from routes.favorites import favorites_bp
11 from routes.photos import photos_bp
12 from routes.promotions import promotions_bp
13 from routes.ratings import ratings_bp
14 from routes.owners import owners_bp
15
16 app = Flask(__name__)
17 CORS(app)
18
19 app.register_blueprint(restaurants_bp, url_prefix="/restaurants")
20 app.register_blueprint(users_bp, url_prefix="/users")
21 app.register_blueprint(reservations_bp, url_prefix="/reservations")
22 app.register_blueprint(recommend_bp, url_prefix="/recommend")
23 app.register_blueprint(menu_bp, url_prefix="/menu")
24 app.register_blueprint(favorites_bp, url_prefix="/favorites")
25 app.register_blueprint(photos_bp, url_prefix="/photos")
26 app.register_blueprint(promotions_bp, url_prefix="/promotions")
27 app.register_blueprint(ratings_bp, url_prefix="/ratings")
28 app.register_blueprint(owners_bp, url_prefix="/owners")
29
30 @app.route("/")
31 def home():
32     return "Smart Restaurant API is running!"
33
34 if __name__ == "__main__":
35     app.run(host="0.0.0.0", port=8000, debug=True)
36
```



But,
Where is
Database
???

Tables in Database

- | | |
|----------------------|---------------------------|
| 1 address | 2 cuisine |
| 3 favorite | 4 menu |
| 5 menuitem | 6 photo |
| 7 promotion | 8 rating |
| 9 reservation | 10 restaurant |
| 11 restaurantcuisine | 12 restaurantopeninghours |
| 13 restauranttable | 14 review |
| 15 useraccount | 16 useractivity |
| 17 useraddress | 18 usersession |

Database Implementation Overview

```
1
2
3 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
4 START TRANSACTION;
5 SET time_zone = "+00:00";
6
7
8 SET foreign_key_checks = 0;
9
10 --
11 -- Database: `smart_restaurant_finder`
12
13 -- Table structure for table `address`
14 --
15 DROP TABLE IF EXISTS `address`;
16
17 CREATE TABLE `address` (
18     `AddressID` int(11) NOT NULL,
19     `StreetAddress` varchar(512) DEFAULT NULL,
20     `Area` varchar(200) DEFAULT NULL,
21     `City` varchar(100) NOT NULL,
22     `State` varchar(100) DEFAULT NULL,
23     `Country` varchar(100) DEFAULT 'Pakistan',
24     `ZipCode` varchar(20) DEFAULT NULL,
25     `Latitude` decimal(10,7) DEFAULT NULL,
26     `Longitude` decimal(10,7) DEFAULT NULL,
27     `CreatedAt` datetime DEFAULT current_timestamp(),
28     `UpdatedAt` datetime DEFAULT current_timestamp() ON UPDATE current_timestamp()
29 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
30
31
32 INSERT INTO `address`(`AddressID`, `StreetAddress`, `Area`, `City`, `State`, `Country`, `Zipcode`, `Latitude`, `Longitude`, `CreatedAt`, `U
33 (4001, 'Street 1', 'Clifton', 'Karachi', 'Sindh', 'Pakistan', '75000', 24.8065320, 67.0362650, '2025-10-26 19:09:51', '2025-10-27 17:11:27')
34 (4002, 'street 2', 'DHA', 'Karachi', 'Sindh', 'Pakistan', '75001', 24.8366370, 67.0946270, '2025-10-26 19:09:51', '2025-10-27 17:11:27'),
```



```
2740
2741 -- Constraints for table `review`
2742 --
2743 ALTER TABLE `review`
2744     ADD CONSTRAINT `fk_review_restaurant` FOREIGN KEY (`RestaurantID`) REFERENCES `restaurant` (`RestaurantID`) ON DELETE CASCADE,
2745     ADD CONSTRAINT `fk_review_user` FOREIGN KEY (`UserID`) REFERENCES `useraccount` (`Id`) ON DELETE CASCADE;
2746
2747 --
2748 -- Constraints for table `useractivity`
2749 --
2750 ALTER TABLE `useractivity`
2751     ADD CONSTRAINT `useractivity_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `useraccount` (`Id`),
2752     ADD CONSTRAINT `useractivity_ibfk_2` FOREIGN KEY (`RestaurantID`) REFERENCES `restaurant` (`RestaurantID`);
2753
2754 --
2755 -- Constraints for table `useraddress`
2756 --
2757 ALTER TABLE `useraddress`
2758     ADD CONSTRAINT `fk_useraddress_address` FOREIGN KEY (`AddressID`) REFERENCES `address` (`AddressID`) ON DELETE CASCADE,
2759     ADD CONSTRAINT `fk_useraddress_user` FOREIGN KEY (`UserID`) REFERENCES `useraccount` (`Id`) ON DELETE CASCADE;
2760
2761 --
2762 -- Constraints for table `usersession`
2763 --
2764 ALTER TABLE `usersession`
2765     ADD CONSTRAINT `usersession_ibfk_1` FOREIGN KEY (`UserID`) REFERENCES `useraccount` (`Id`) ON DELETE CASCADE;
2766 COMMIT;
2767 SET FOREIGN_KEY_CHECKS = 1;
```



Database hosted online on AIVEN + MYSQL Workbench

The screenshot shows the AIVEN Console interface. In the top navigation bar, there are links for Home, Projects, Tools, Billing, Support, and Admin. The main area is titled "My Organization / smart-restaurant-finder / Services". A prominent blue button labeled "Create service" is visible. On the left, a sidebar lists various project components: Get started, Services (which is selected), Integration endpoints, VPCs, Event log, Permissions, Project AI insights, and Settings. The main content area displays a table with columns: Service, Nodes, Plan, Cloud, Created, and Action. One service entry is shown: "mysql-Ba37a39" (MySQL, Running, 2 nodes, Business-4 plan, Google Cloud: asia-south2, Asia, India, created 4 days ago). A search bar and a "Filter list" button are also present.

The screenshot shows the MySQL Workbench interface. The title bar says "MySQL Workbench aiven-smart-restaurant". The left pane is the Navigator, showing the "defaultdb" schema with tables like address, cuisine, favorite, menu, etc. The central pane contains a query editor with the SQL command "SELECT * FROM defaultdb.cuisine;". Below it is a "Result Grid" showing the results of the query. The results are as follows:

CuisineID	CuisineType	Description	CreatedAt
8	Seafood	Seafood cuisine	2025-10-26 19:09:50
9	Desserts	Desserts cuisine	2025-10-26 19:09:50
10	Middle Eastern	Middle Eastern cuisine	2025-10-26 19:09:50
11	Japanese	Japanese cuisine	2025-10-26 19:09:50
12	Thai	Thai cuisine	2025-10-26 19:09:50
13	Mexican	Mexican cuisine	2025-10-26 19:09:50
14	Indian	Indian cuisine	2025-10-26 19:09:50
15	Mediterranean	Mediterranean cuisine	2025-10-26 19:09:50
16	Lebanese	Lebanese cuisine	2025-10-26 19:09:50
17	Korean	Korean cuisine	2025-10-26 19:09:50

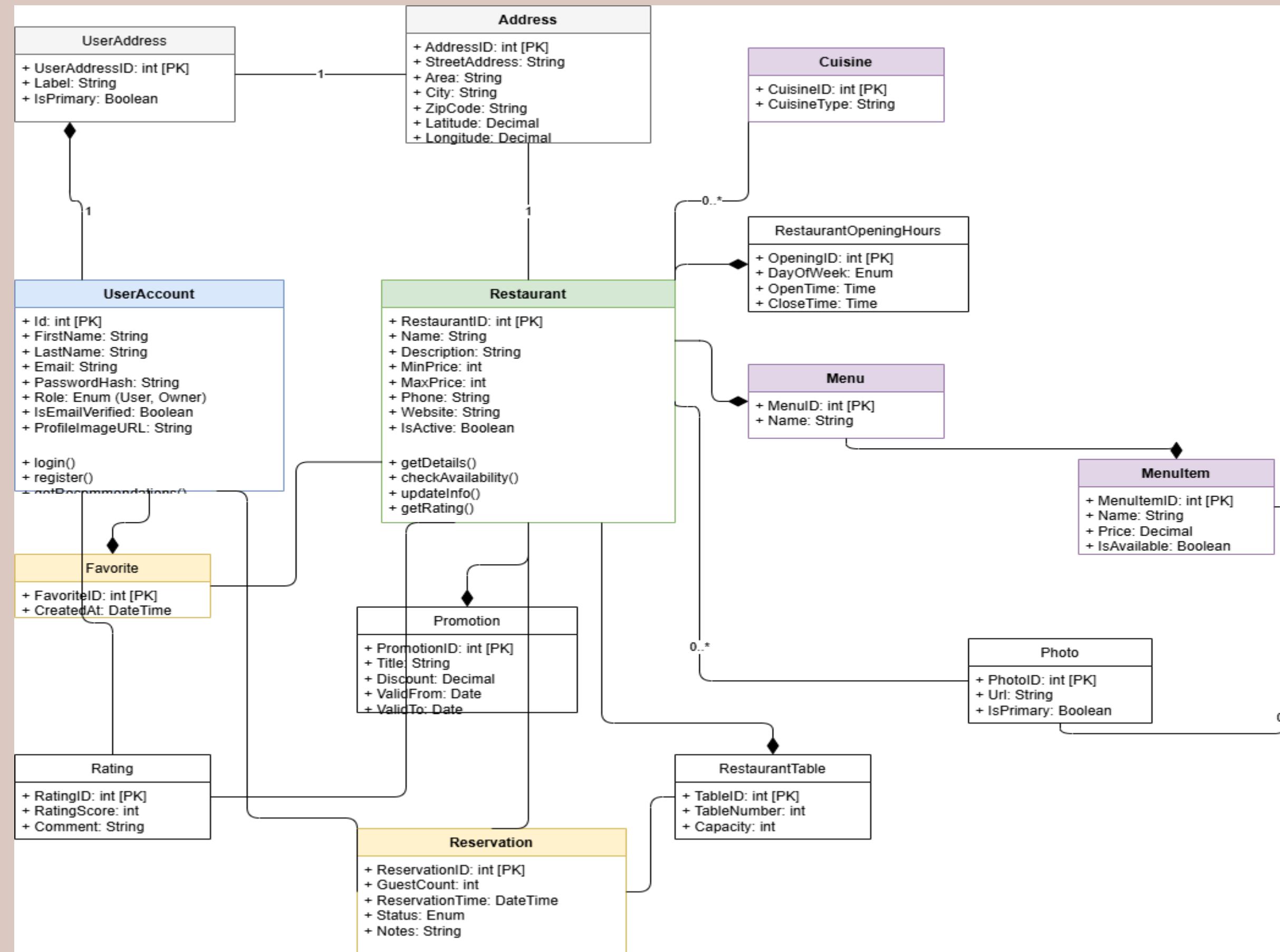
The right pane contains a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." The bottom status bar shows the duration of the fetch operation: "0.203 sec / 0.000 sec".



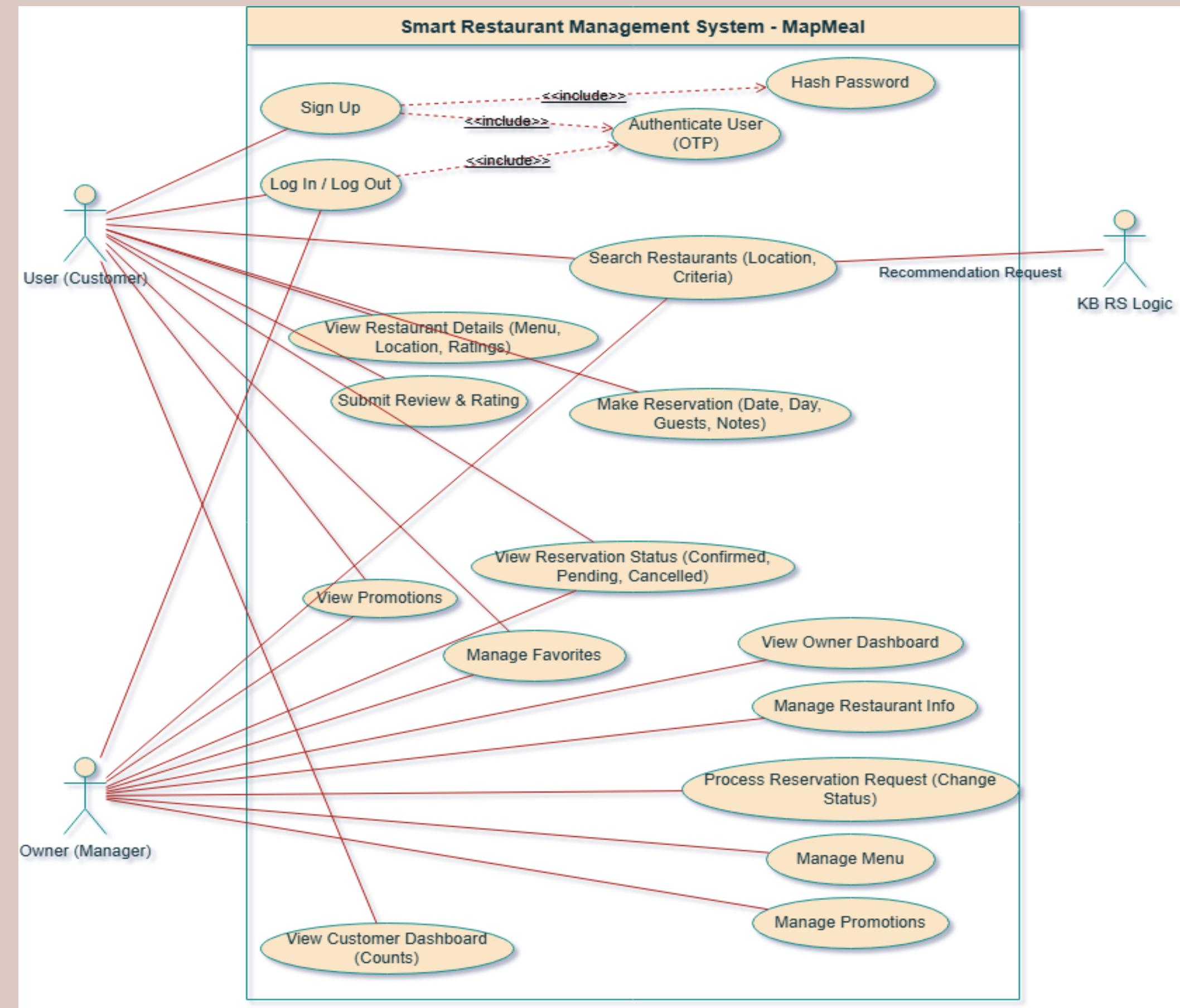
Moving Towards Diagrams
to understand the actual
flow of Meal-Map



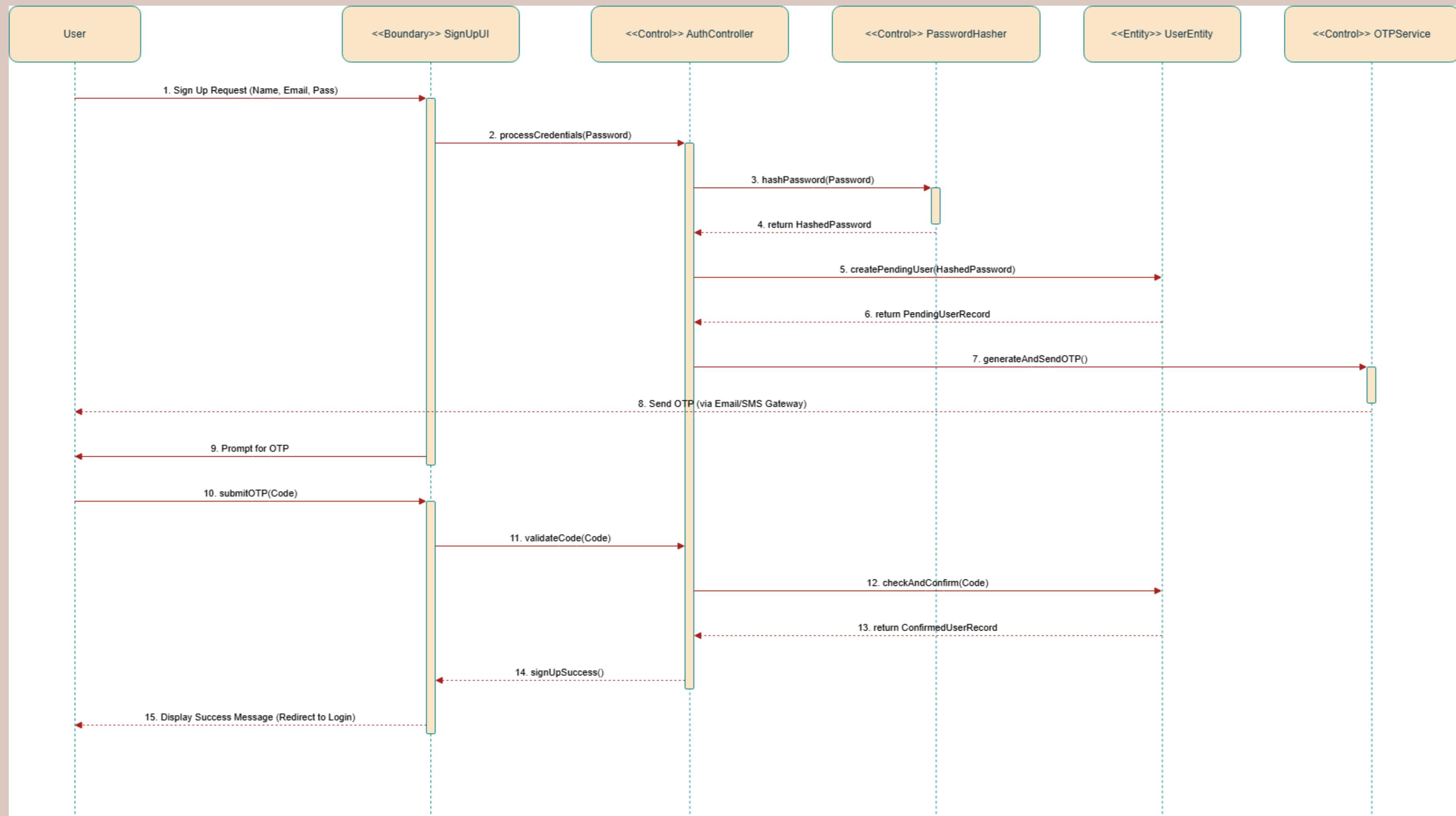
Class Diagram



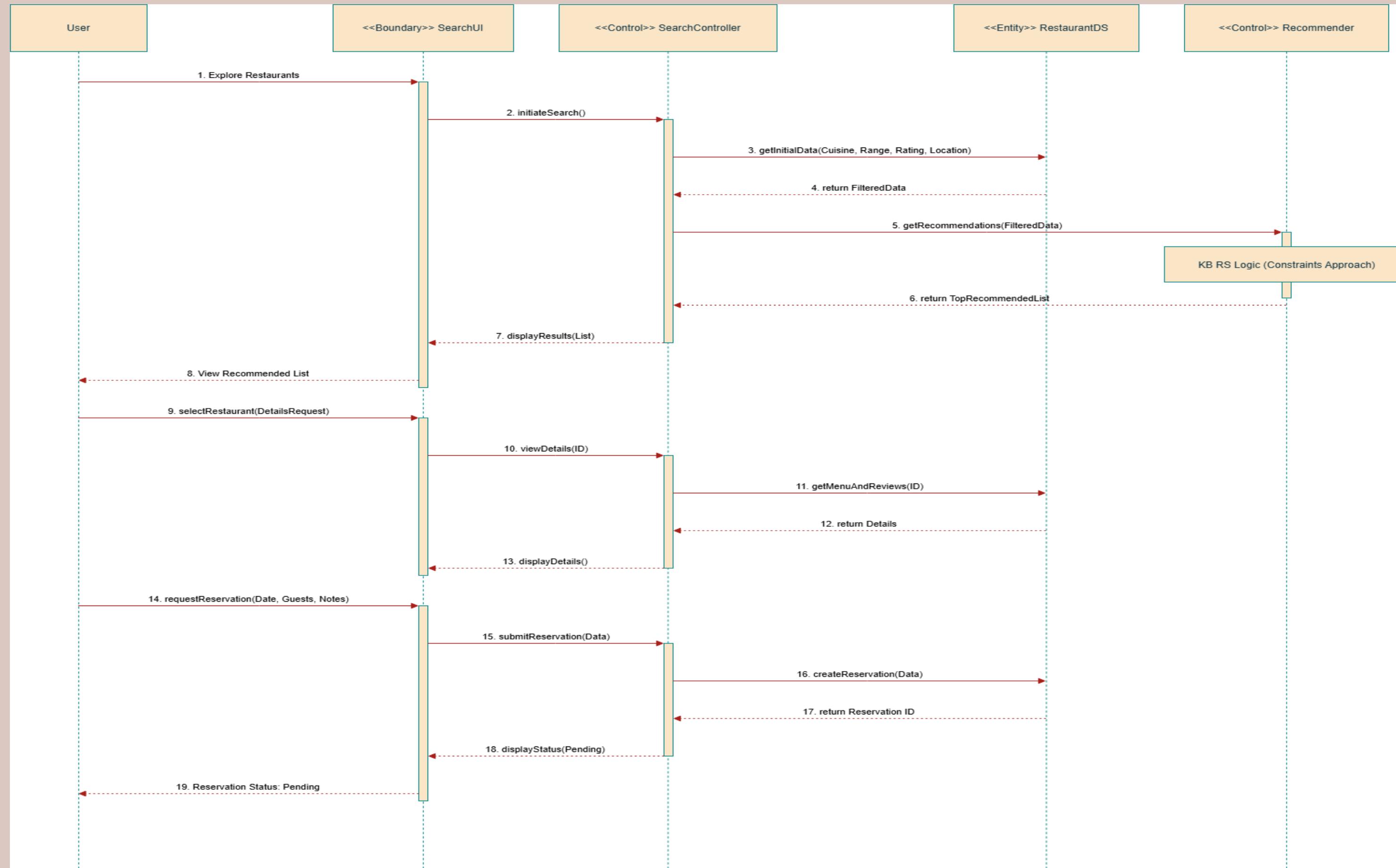
Use-Case Diagram



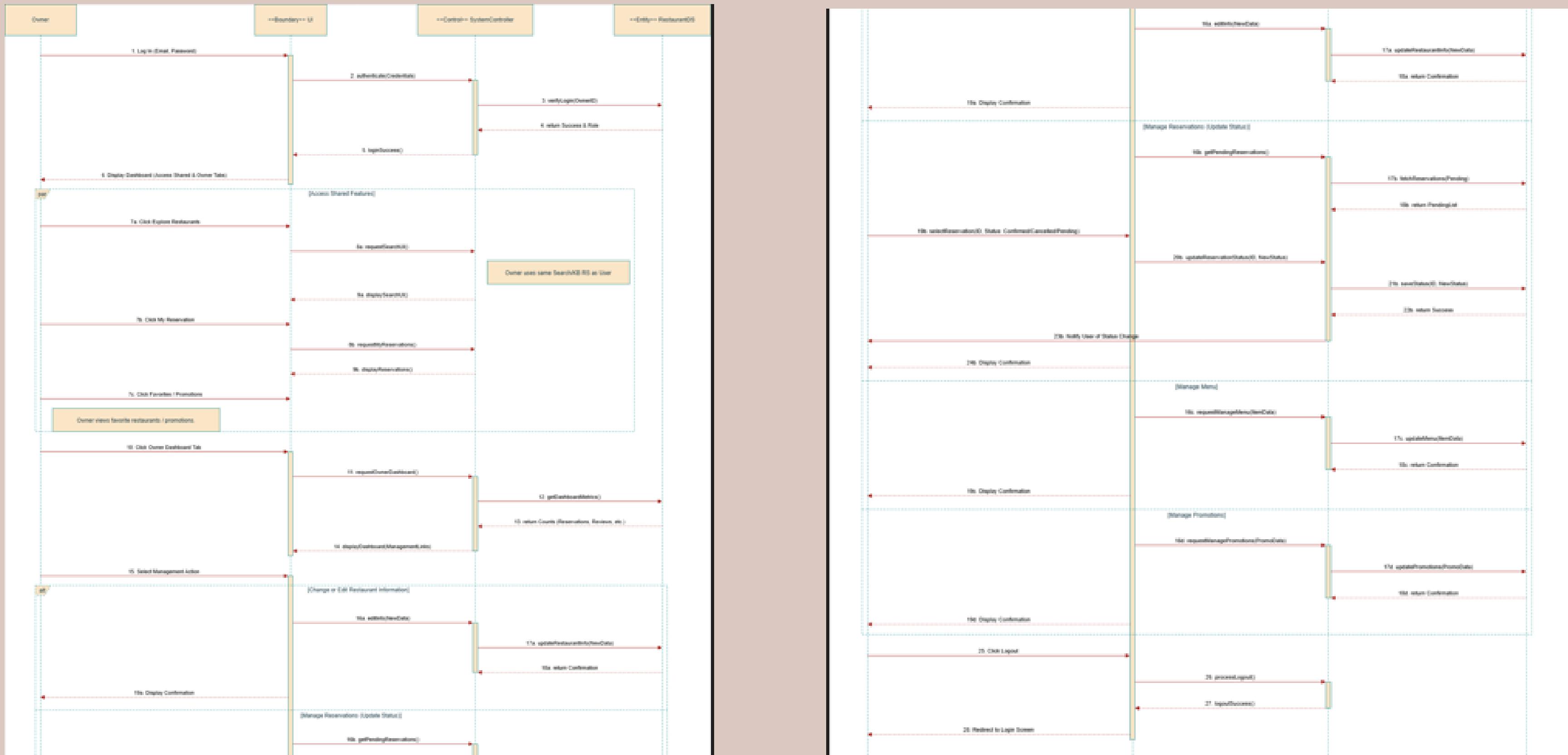
Sequence Diagram for Sign-up



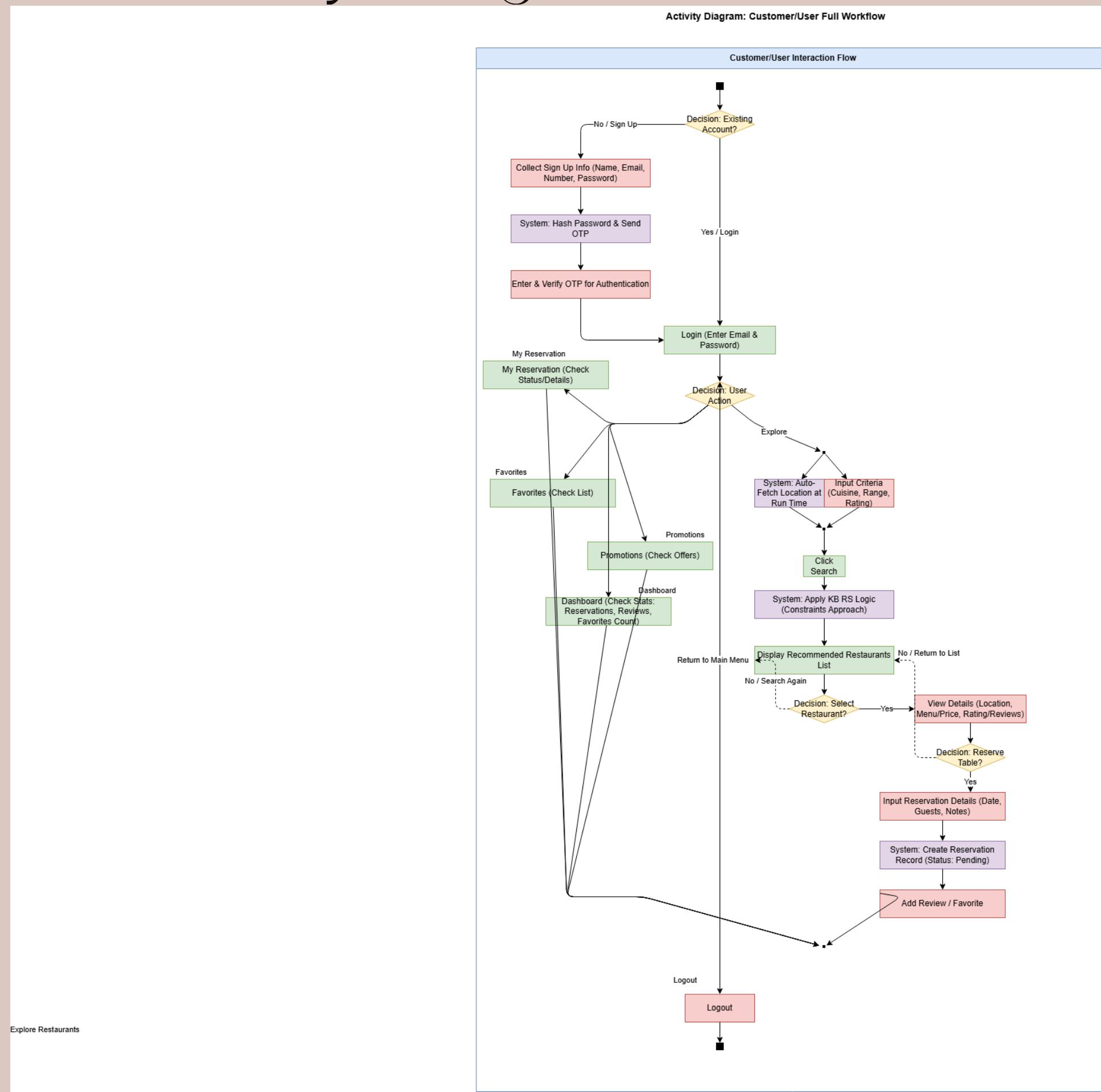
Sequence Diagram for User



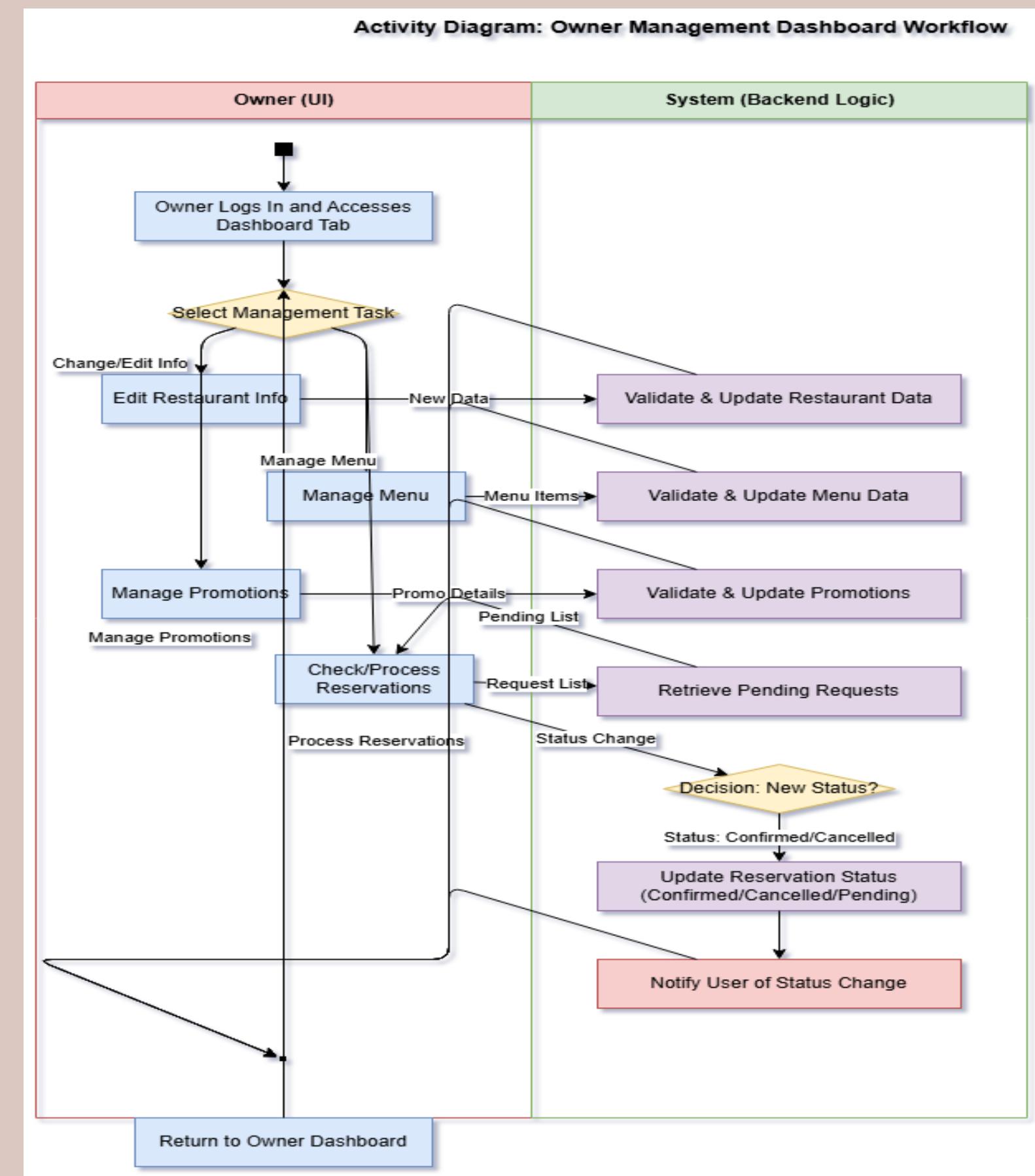
Sequence Diagram for Owner



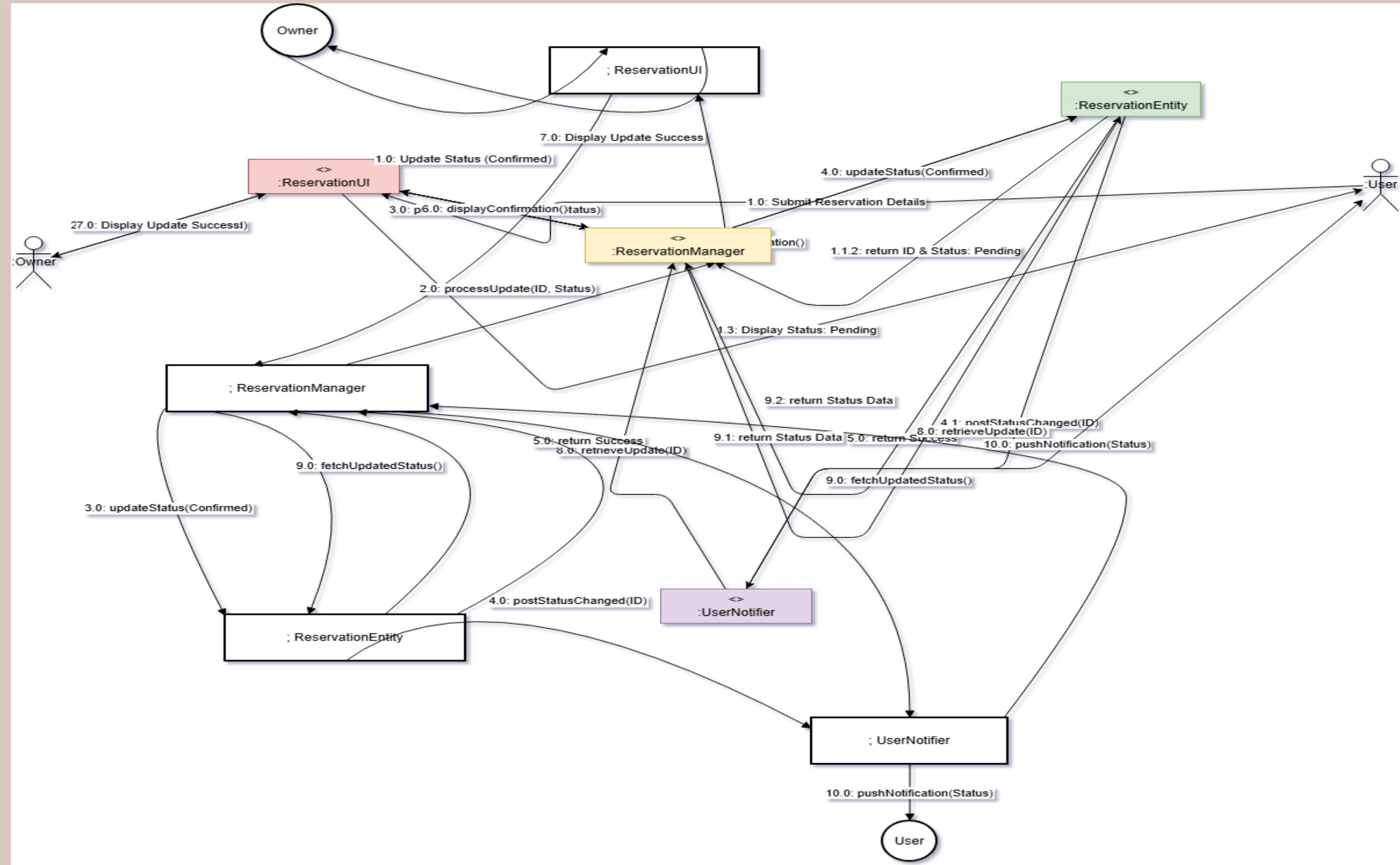
Activity Diagram for User



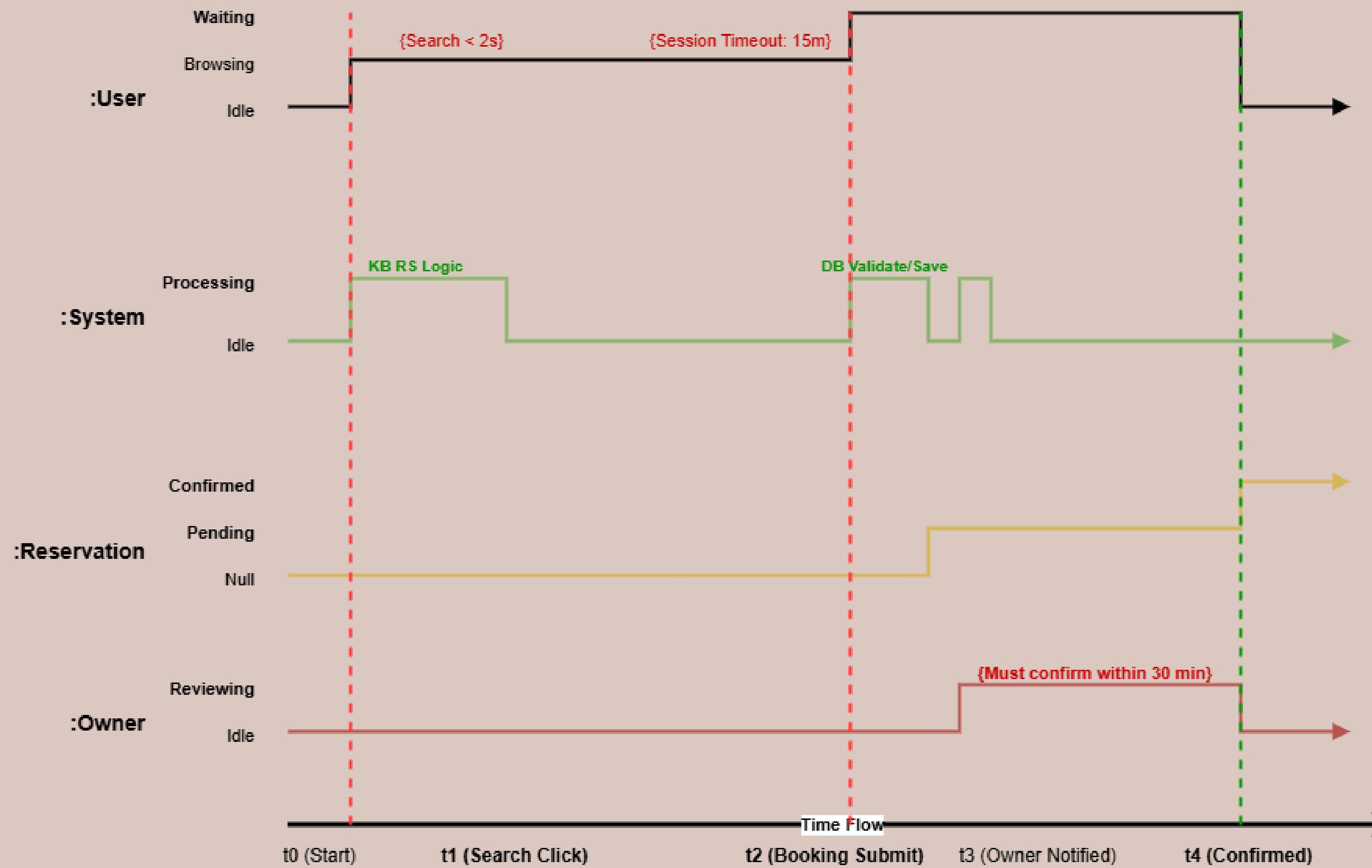
Activity Diagram for Owner



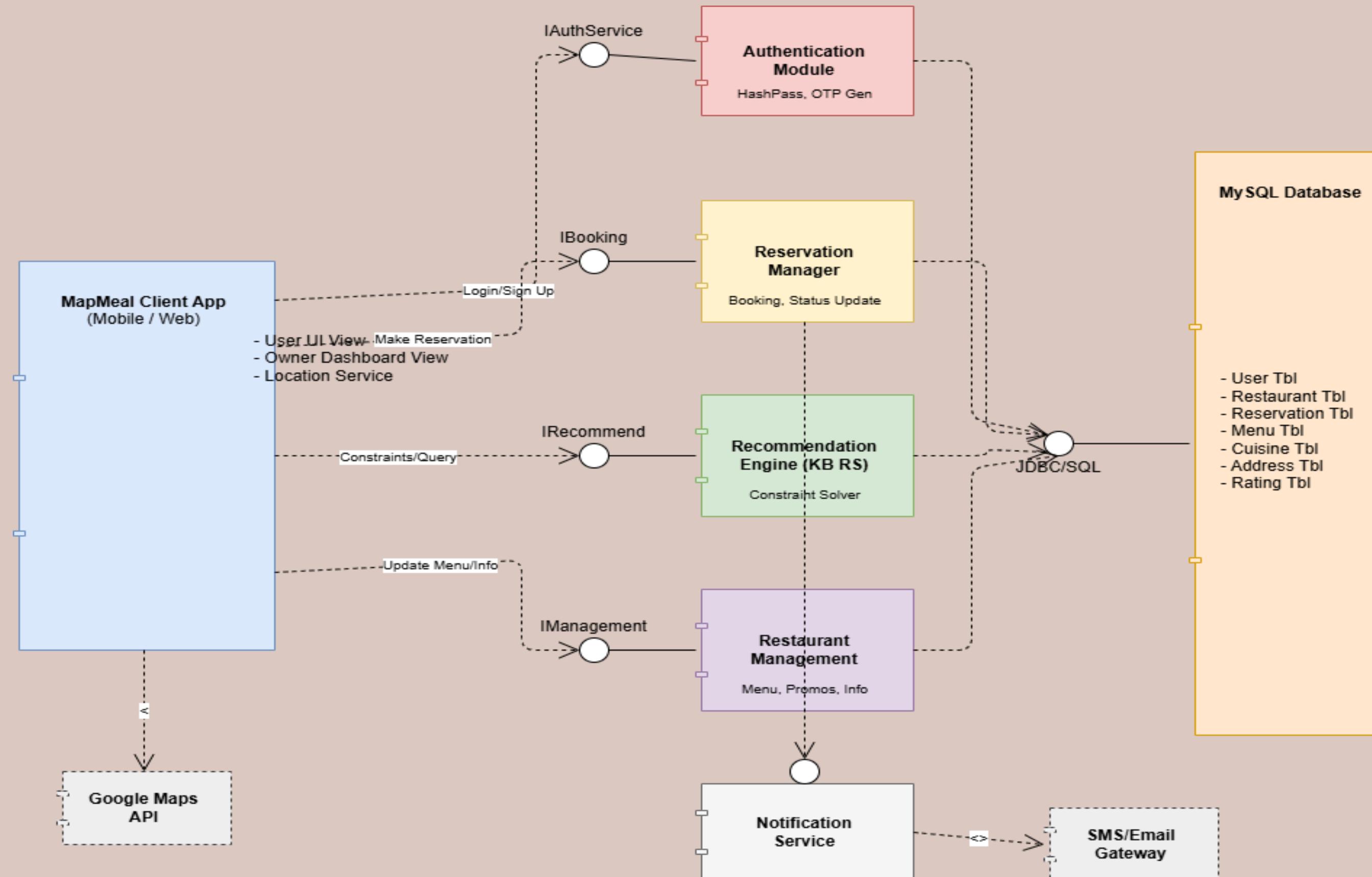
Collaborative Diagram



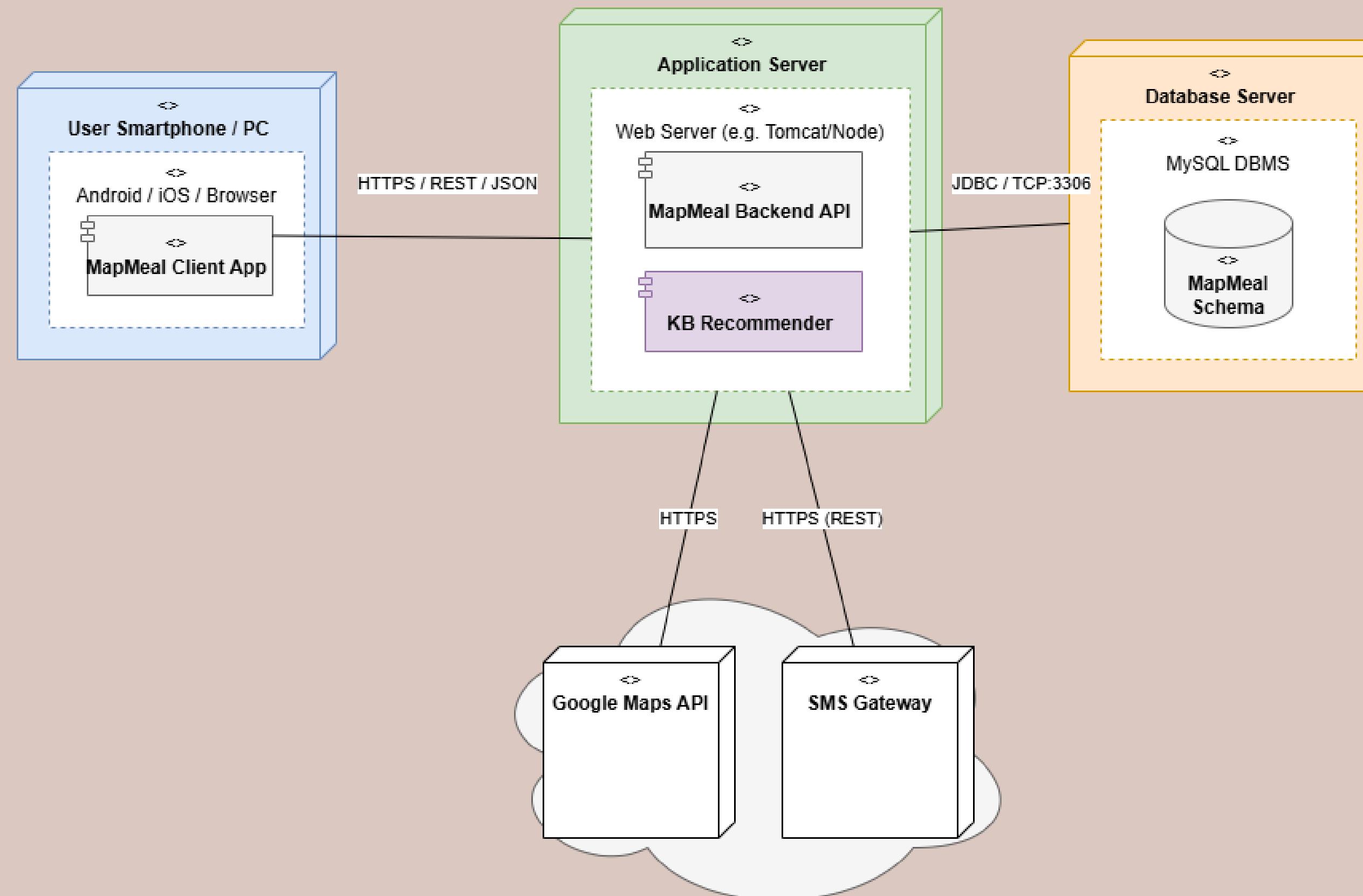
Timing Diagram



Component Diagram



Deployment Diagram



03.

RS Implementation



A photograph of a woman with curly hair, wearing a grey blazer over a white shirt, smiling and holding a piece of sushi with chopsticks. She is wearing a gold chain necklace and a gold watch. The background is dark.

Wanna Know
Which RS
Technique is used
in this Project?

It's
Knowledge Base
Recommendation
System





★ RS - Overview

Type: Knowledge-Based Recommender System (KB-RS)

Why KB-RS?

- Suitable where the historical data is small
- Predictions based on rules & constraints
- Works even for new restaurants



KB-RS Constraints Approach

User Preferences:

- Cuisine
- Maximum Budget
- Minimum Budget
- Location (auto fetched at run time)

System Filters:

- Only restaurants matching constraints
- Rank them by score
- Recommend Top Restaurants

KB-RS Constraints

1. Preference Constraint (Cuisine Constraint)

The system filters restaurants **based on explicit user preferences**.

Only restaurants whose cuisine belongs to the **user-defined preference set** are considered.

2. Quality Threshold Constraint (Rating Constraint)

A **minimum acceptable quality threshold** is enforced.

Restaurants are included only if their **rating \geq user-defined rating threshold**.

3. Budget Constraint (Price Constraint)

A **cost-based constraint** ensures recommendations remain within the user's budget.

Restaurants must satisfy:

average_price \geq minimum acceptable price level.

KB-RS Constraints

4. Spatial/Contextual Constraint (Location Constraint)

A **context-aware constraint** using the user's current or selected location.

Only restaurants in the **relevant spatial context** (location/area) are recommended.

5. Ordering Constraint (Sorting by Utility Function)

After filtering, restaurants are ranked using a **utility-based ordering rule**, e.g.:

Utility = rating (default)

Results sorted by the selected **utility dimension**.

6. Cardinality Constraint (Limit Constraint)

A **cardinality restriction** limits the size of the recommendation set.

System returns only the **top-N most relevant items** (default N = 30).

KB-RS Implementation in Meal-Map

KB-RS Implementation in Meal-Map

```
# Rating filter
if min_rating is not None:
    sql += " AND r.AvgRating >= %s"
    params.append(min_rating)

# LOCATION FILTER
# LOCATION FILTER (partial match)
if location:
    sql += " AND a.Area LIKE %s"
    params.append(f"%{location}%") # adds wildcard before and after

sql += " GROUP BY r.RestaurantID, r.Name, r.PriceRange, rt.RatingScore, a.Area"

# Sorting
if sort_by == "rating":
    sql += " ORDER BY AvgRating DESC, TotalRatings DESC"
elif sort_by == "price_asc":
    sql += " ORDER BY r.minPrice ASC"
elif sort_by == "price_desc":
    sql += " ORDER BY r.maxPrice DESC"
elif sort_by == "totalratings":
    sql += " ORDER BY TotalRatings DESC"
else:
    sql += " ORDER BY AvgRating DESC"

sql += " LIMIT %s"
params.append(limit)

conn = get_db()
cursor = conn.cursor(buffered=True)
cursor.execute(sql, tuple(params))
records = cursor.fetchall()
cursor.close()
conn.close()
```

```
results = []
print('records: ', records)
for row in records:
    rid, name, price, avg, total, area, cuisines_str = row

    results.append([
        "RestaurantID": rid,
        "Name": name,
        "PriceRange": float(price) if price is not None else None,
        "AvgRating": float(avg),
        "TotalRatings": int(total),
        "Area": area,
        "Cuisines": cuisines_str or ""
    ])

return results
```

Conclusion



Complete smart restaurant system
Fully integrated SDA + RS + DB
Real-time recommendations
Complete user & owner flows

Created by

Syed Waleed Hussain
23K-0885

Sofia Ayaz
23K-0807

Shayan Nemat
23K-0899





Thank you!

Enjoy your Meal-Map